

# [SE3BB4] Software Design III: Concurrent System Design

## Assignment 2

Due November 16, 2015 FSP MODEL

Due November 23, 2015 Java Implementation

**Instructions:** Please read the questions carefully. Solutions must be submitted in the Avenue Dropbox created for the assignment. Note that the deadline is strictly enforced. The avenue system tracks the exact time that submissions are uploaded and late submissions may be rejected.

### Question 1

Consider the following problem statement:

Assume that in order to have a smoke a smoker requires: **paper**, **tobacco** (items needed to make a cigarette), and a **spark** (item needed to light the cigarette). Further, assume there are three smokers, called **HORACIO**, **ARTHUR**, **EDGAR**, seated at a round table such that: **HORACIO** has an infinite supply of **tobacco**, **ARTHUR** has an infinite supply of **paper**, and **EDGAR** has an infinite supply of matches.

The problem statement reads as follows: If a smoker wants to have a smoke, he must obtain (arbitrarily) from the table the two remaining ingredients needed (e.g., if **HORACIO** wants to smoke, he must obtain a piece of paper and a match). It is only after these two remaining ingredients have been obtained that the smoker can proceed to smoke. Smokers smoke continuously.

The following constraints are in order: (1) Done by an agent called **AGATHA**, ingredients must be place on the table before a smoker can obtain them; (2) smokers do not hoard any of the ingredients; (3) a smoker only begins to smoke a new cigarette once he has stubbed out the last one; (4) smokers cannot be interrupted or stopped at anytime.

Included as part of this assignment, the file `cigarette.smokers.lts` contains a possible way of modeling this problem in FSP. However, as is easy to see resorting to the LTSA tool, this model possesses some problems. Your task is to solve these problems following the ideas proposed by David Parnas in [1]. More so, you should guarantee that the FSP model you propose is deadlock free and that every smoker smokes infinitely many times.

### Implementation hints

As part of your Java implementation you should consider:

- (a) Carefully documenting any assumptions and design decisions you make.
- (b) Creating a Makefile and a run script – further details can be sought during the tutorials.
- (c) Using Java library classes whenever possible.
- (d) Designing, document, and testing your code – provide a short report detailing these points.

### Understanding the problem and discussing possible solutions

You are strongly urged to ask questions during the tutorials, both for clarifications on the problem and for discussing possible solutions. Also, you may discuss how to approach the assignment with your fellow classmates. If you do that, you must include a list of all those with whom you discussed the exercise in your final submission and describe

what information or hints you received from them. It is part of your professional responsibility to give credit to all who have contributed to your product. This said, your final submission must be your group's own work. If it is discovered that you have not written your own solutions, or that you have consulted with people not mentioned in your submission, it will be considered an act of academic dishonesty.

## References

- [1] Parnas, D. L.. *On a solution to the cigarette smoker's problem*. Communications of the ACM. Vol 18, Issue 3. March 1975.